

Penerapan Metode String Matching dengan Algoritma Rabin Karp untuk Merekomendasikan Objek Wisata di Kabupaten Bulukumba Berbasis Web

Hadi Yusuf^{a,1,*}, Ramdan Satra^{a,2}, Farniwati Fattah^{a,3}

^a Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muslim Indonesia, Jl. Urip Sumoharjo KM. 05, Makassar dan 90231, Indonesia^b

¹ hadiyusuf889@gmail.com; ² ramdan@umi.ac.id; ³ farniwati.fattah@umi.ac.id;
*corresponding author

INFORMASI ARTIKEL	ABSTRAK
Diterima : 24 – 07 – 2023 Direvisi : 01 – 11 – 2023 Diterbitkan : 04 – 12 – 2023	Pariwisata merupakan salah satu kegiatan yang digemari oleh banyak kalangan saat ini, bahkan dapat dikatakan bahwa pariwisata merupakan salah satu kebutuhan yang penting bagi orang banyak, terutama menyangkut kegiatan sosial ekonomi yang dipandang sebagai salah satu industry pariwisata yang prospektif di masa yang akan datang. Pariwisata yang saat ini menarik diberbagai kalangan, mendorong pengembangan sektor pariwisata, salah satunya pengembangan dengan menggunakan teknologi digital. Pemanfaatan teknologi digital dan internet merupakan salah satu upaya dalam mengembangkan pariwisata baik dalam kegiatan promosi atau kegiatan dalam menjalankan bisnis pariwisata. Salah satu bidang ilmu dalam Ilmu Komputer yang dapat dimanfaatkan pada segmentasi pariwisata adalah metode <i>String Matching</i> untuk proses pencarian objek wisata. Metode pencocokan <i>String</i> atau <i>String Matching</i> ini dapat digunakan dengan menggunakan beberapa Algoritma, salah satunya adalah <i>Rabin-Karp</i> . Pada penelitian ini, peneliti memanfaatkan metode <i>String Matching</i> dalam pencarian objek wisata berdasarkan deskripsi kriteria tempat wisatanya yang berupa data objek wisata berupa nama, lokasi, sarana dan prasana, alamat google maps, gambar, dan kriteria objek wisata lainnya yang diinputkan. Hasil dari penelitian ini adalah sebuah sistem yang menerapkan Algoritma <i>Rabin-Karp</i> untuk merekomendasikan objek wisata di Kabupaten Bulukumba yang dimana persentase kemiripan antara inputan pencarian objek wisata dengan hasil rekomendasi wisata yaitu sebesar 33.49%.
Kata Kunci: Pariwisata <i>String Matching</i> <i>Rabin-Karp</i>	
	This is an open access article under the CC-BY-SA license
	

I. Pendahuluan

Kabupaten Bulukumba adalah salah satu kabupaten pesisir di Sulawesi Selatan yang memiliki sejarah dan budaya masyarakat yang kaya dengan khazanah kehidupan peisisir dan laut[1]. Kabupaten Bulukumba ini menjadi salah satu daerah atau kabupaten di Sulawesi Selatan dengan objek wisata yang banyak. Daerah Bulukumba ini merupakan daerah yang direkomendasikan untuk dikunjungi saat berada di Sulawesi Selatan.

Sumber informasi terkait pariwisata di Kabupaten Bulukumba dapat dilihat dari beberapa system informasi daerah, dan aplikasi pariwisata seperti aplikasi ticketing dan lainnya. Wisatawan yang hendak berkunjung ke Bulukumba untuk menikmati objek wisata disana, mendapatkan informasi dari internet, atau informasi dari kenalan, teman, atau masyarakat. Kebutuhan informasi objek wisata yang dibutuhkan wisatawan sangat penting karena kriteria objek wisata yang ingin diketahui wisatawan sebelum berkunjung. Kegiatan yang wisatawan lakukan untuk mendapatkan informasi objek wisata dengan melihat kriteria objek wisata dilakukan dengan membaca atau mencari informasi objek wisata menggunakan sistem informasi yang tersedia pada sistem informasi daerah dan aplikasi pariwisata dan hal ini kurang maksimal bagi wisatawan untuk mendapatkan rekomendasi tempat wisata dengan kriteria yang diinginkan oleh wisatawan.

Salah satu bidang ilmu dalam Ilmu Komputer yang dapat dimanfaatkan pada segmentasi pariwisata adalah metode *String Matching* untuk proses pencarian objek wisata. *String Matching* adalah proses pencarian sebuah *String* yang terdiri dari pattern terhadap karakter pada teks yang dicari[2]. Metode pencocokan *String* atau *String Matching* ini dapat digunakan dengan menggunakan beberapa Algoritma, salah satunya adalah *Rabin-Karp*. *Rabin-Karp* merupakan metode pencocokan *String* dengan menggunakan fungsi hashing untuk menemukan *pattern* atau pola di dalam teks[3]. Penelitian yang telah memanfaatkan *String Matching* dalam

proses pencarian objek wisata adalah penelitian Ade Ambarwati Br Ginting dan Dito Putro Utomo dengan judul Perancangan Aplikasi Catalog Wisata di Sumatera Utara Menggunakan Algoritma Rabin Karp. Pada penelitian tersebut, pencarian objek wisata dengan pengguna memasukkan kata kunci objek wisata untuk menampilkan objek wisata yang dicari oleh pengguna.

Pada penelitian ini, peneliti memanfaatkan metode *String Matching* dalam pencarian objek wisata berdasarkan deskripsi kriteria tempat wisatanya yang berupa data objek wisata berupa nama, lokasi, sarana dan prasana, alamat google maps, gambar, dan kriteria objek wisata lainnya yang diinputkan. Metode *String Matching* akan digunakan untuk mencocokkan deskripsi kriteria objek wisata dengan kriteria objek wisata yang diinputkan oleh wisatawan. Dengan memanfaatkan algoritma *Rabin-Karp*, wisatawan akan mendapatkan rekomendasi objek wisata yang sesuai dengan deskripsi kriteria objek wisata yang diinputkan oleh wisatawan dengan membandingkan pola teks dari kriteria masing-masing objek wisata dengan teks kriteria objek wisata yang dicari oleh wisatawan. Berdasarkan hal ini, maka pada penelitian ini, metode tersebut sebagai alat pencarian objek wisata berdasarkan kriteria khususnya pada objek wisata di Kabupaten Bulukumba.

II. Metode

a. *String Matching*

String Matching menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST)* adalah susunan dari karakter-karakter (angka, alfabet atau karakter yang lain) dan biasanya direpresentasikan sebagai struktur dan array. *String Matching* atau pencocokan string adalah suatu metode untuk menemukan suatu keakuratan/hasil dari beberapa pola teks yang diberikan[4].

String Matching adalah sebuah metode yang digunakan dalam sebuah pencocokkan suatu pola kata tertentu terhadap suatu kalimat atau teks Panjang. Metode *String Matching* sendiri dapat dilakukan dengan beberapa algoritma tertentu, antara lain *Brute Force*, *Knuth Morris Pratt (KMP)* dan *Rabin-Karp*[5].

Penggunaan *String Matching* mencakup pencarian pola dalam *DNA sequence*, *search engine* internet, menemukan halaman web yang relevan pada *query*, dapat pula dimanfaatkan untuk mendeteksi adanya plagiarisme karya tulis. Termasuk dalam algoritma string matching diantaranya Algoritma Naïve, algoritma *Rabin-Karp*, algoritma *Finite Automaton* dan Algoritma *Knuth Morris Pratt*. *String* dalam ilmu computer dapat diartikan dengan sekuens dan karakter. Walaupun sering juga dianggap sebagai data abstrak yang menyimpan sekuens nilai data, atau biasanya berupa bytes yang mana merupakan elemen yang digunakan sebagai pembentuk karakter sesuai dengan encoding karakter yang disepakati seperti ASCII, ataupun EBCDIC. Pencocokan string atau *String Matching* adalah proses pencarian semua kemunculan string pendek $P[0...n-1]$ yang disebut *pattern* di string yang lebih Panjang $T[0...m-1]$ yang disebut teks. Pencocokan string merupakan permasalahan paling sederhana dari semua permasalahan string lainnya, dan merupakan bagian dari pemrosesan data, pengkompresian data, *lexical analysis*, dan temu balik informasi. Teknik untuk menyelesaikan permasalahan pencocokan string biasanya akan menghasilkan implikasi langsung ke aplikasi string lainnya[4].

b. Algoritma Rabin-Karp

Algoritma *Rabin-Karp* diciptakan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 yang menggunakan fungsi *hashing* untuk menemukan *pattern* dalam *string* teks. Charas and Lecroq, menyatakan bahwa fungsi *hashing* menyediakan metode sederhana untuk menghindari perbandingan jumlah karakter yang kuadratik di dalam banyak kasus atau situasi. Dari pada melakukan pemeriksaan terhadap setiap posisi dari teks Ketika terjadi pencocokan pola, akan lebih baik dan efisien melakukan pemeriksaan hanya jika teks yang sedang proses memiliki kemiripan seperti pada *pattern*[6]. Untuk melakukan pengecekan kemiripan antara dua kata ini digunakan fungsi hash. Algoritma ini menggunakan fungsi hash dalam mencari suatu kata. Teori ini jarang digunakan untuk mencari kata tunggal, namun cukup penting dan sangat efektif untuk pencarian *pattern*[7].

Ade Ambarwati br Ginting dan Dito Putro Utomo melakukan penelitian dengan judul Perancangan Aplikasi Catalog Wisata di Sumatera Utara Menggunakan Algoritma *Rabin-Karp*, dengan hasil penelitian sangat membantu wisatawan karena dapat melakukan pencarian *string* dengan cepat dan memberikan hasil yang tepat. Adapun perbedaan dengan penelitian yang diusulkan yaitu menggunakan objek penelitian yang berbeda dan pada penelitian yang diusulkan akan mendapatkan rekomendasi objek wisata berdasarkan dengan kriteria yang diinputkan[8].

Eric Siswanto dan Yo Ceng Giap melakukan penelitian dengan judul penelitian Implementasi Algoritma *Rabin-Karp* dan *Cosine Similarity* untuk pendeteksi plagiarisme pada dokumen dengan hasil perbandingan dokumen satu dan dokumen kedua sebesar 32%. Adapun perbedaan dengan penelitian yang diusulkan yaitu cuman menggunakan metode *Rabin-Karp* dan implementasi metode *Rabin-Karp* pada pencarian objek wisata[9].

Desi Novita Sari dan Dito Putro Utomo melakukan penelitian dengan judul Implementasi Algoritma *Rabin-Karp* pada Pencarian Quotes Tokoh Nasional, dengan hasil penelitian algoritma *Rabin-Karp* melakukan pemeriksaan terhadap setiap posisi dari teks Ketika terjadi pencocokan pola, akan lebih efisien melakukan pemeriksaan hanya jika teks yang sedang diproses memiliki kemiripan. Adapun perbedaan

dengan penelitian yang diusulkan yaitu menggunakan objek penelitian yang berbeda dan penelitian terhadap hanya memiliki satu kriteria yaitu Quotes[6].

Algoritma *Rabin-Karp* memiliki beberapa karakteristik yaitu menggunakan *K-Gram* dan *hashing*. Penerapan algoritma *Rabin-Karp* 17 dilakukan setelah melewati tahapan *preprocessing*. Berikut tahapan algoritma *Rabin-Karp*[3]:

- K-Gram* adalah rangkaian token (*Term*) yang Panjang dengan Panjang *k*. metode *K-Gram* mengambil potongan-potongan karakter huruf sejumlah nilai *k* dari sebuah teks yang secara kontinuitas dibaca dari awal teks sumber hingga akhir teks sumber.
- Hashing*. *Hashing* merupakan salah satu cara untuk mengubah karakter *string* pada setiap *Term* hasil *K-Gram* menjadi *integer* yang disebut nilai hash. Proses pengubahan menjadi nilai hash menggunakan fungsi rolling hash. Persamaan rolling hash dapat dilihat pada persamaan

$$H(c_1 \dots c_k) = (c_1 \cdot b^{(k-1)} + c_2 \cdot b^{(k-2)} + \dots + c_k) \bmod q \quad (1)$$

Keterangan:

H : substring

c : nilai ASCII per-karakter

b : konstan bilangan prima

k : banyak karakter

q : module bilangan prima

- Menghitung Similarity. Menghitung Similarity adalah tahapan menghitung nilai similaritas. Proses penghitungan similaritas menggunakan persamaan berikut.

$$S = ((2 * C) / (A + B)) * 100 \quad (2)$$

Keterangan:

S : Nilai Similaritas

A : jumlah Term pada dokumen A

B : jumlah Term pada dokumen B

C : jumlah Term yang sama dari A dan B

III. Hasil dan Pembahasan

Hasil dari penelitian ini adalah sebuah sistem perkomendasi objek wisata di Kabupaten Bulukumba yang dilakukan menggunakan metode algoritma *Rabin-Karp*. Bagian hasil penelitian ini yaitu hasil sistem yang berhasil dibuat dengan mengimplementasikan algoritma *Rabin-Karp*.

A. Hasil

1. Halaman data objek wisata

Modul pengelolaan objek wisata adalah modul untuk menampilkan data objek wisata. Berikut tampilan umum dari modul pengelolaan objek wisata.



Gambar 1. Halaman Pengelolaan Data Objek Wisata

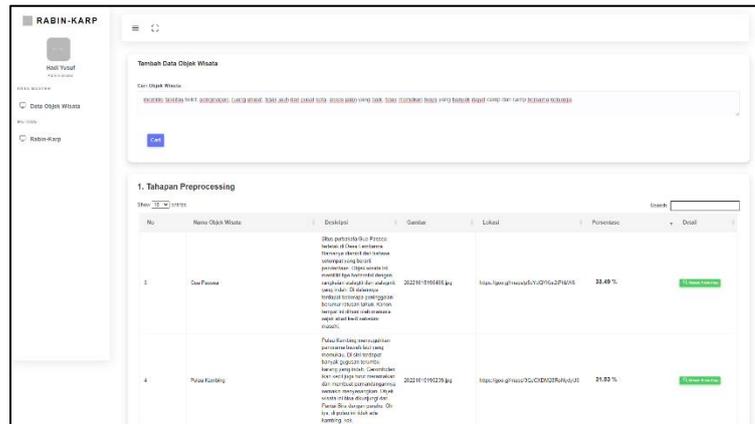
2. Modul perkomendasi objek wisata

Modul perkomendasi objek wisata adalah modul untuk melakukan pengecekan kemiripan objek wisata yang dimasukkan pengguna dan menampilkan hasil pengecekan kemiripan data objek wisata. Berikut bentuk form untuk melakukan perkomendasi objek wisata.



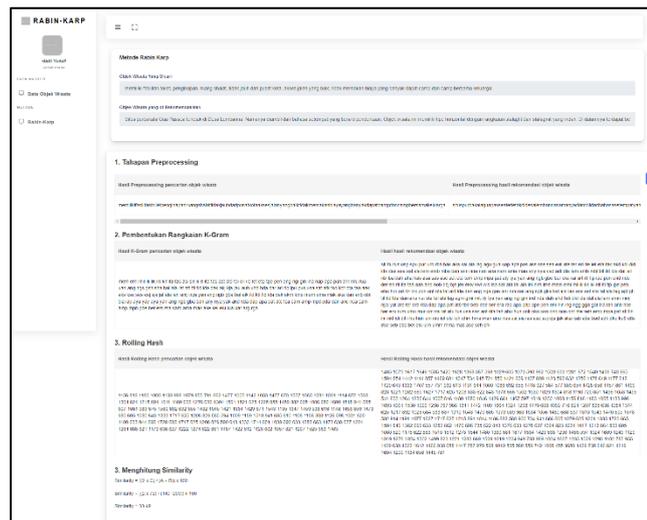
Gambar 2. Form perkomendasi objek wisata

Setelah melengkapi form untuk melakukan rekomendasi objek wisata, maka hasil yang ditampilkan adalah data objek wisata yang disertai dengan nilai kemiripan masing-masing objek wisata dengan pencarian yang diinputkan oleh pengguna. Berikut tampilan untuk hasil rekomendasi objek wisata.



Gambar 3. Hasil rekomendasi objek wisata

Pada sistem yang dibangun, karena penulis berfokus pada pengimplementasian metode algoritma Rabin-Karp, maka penulis menampilkan proses metode algoritma Rabin-Karp dalam melakukan rekomendasi objek wisata. Berikut tampilan untuk menampilkan proses dari metode algoritma Rabin-Karp.



Gambar 4. Proses pengecekan metode algoritma Rabin-Karp

B. Pembahasan

Penelitian ini berokus pada pengimplementasian metode algoritma Rabin-Karp untuk merekomendasikan objek wisata. Pada bagian pembahasan ini akan membahas yaitu bagaimana alur dari metode algoritma Rabin-Karp dalam melakukan pengecekan kemiripan hasil pencarian objek wisata berdasarkan percobaan yang telah dilakukan dan dipaparkan pada bagian hasil penelitian.

Pada pembahasan pengimplementasian metode algoritma *Rabin-Karp* akan menggunakan nilai Variabel *K-Gram* = 3, *b* = 7, dan *mod* = 5001[10]. Adapun 2 teks yang akan di uji adalah sebagai berikut.

Objek wisata yang dicari: “memiliki fasilitas toilet, penginapan, ruang shalat, tidak jauh dari pusat kota, akses jalan yang baik, tidak memakan biaya yang banyak dapat camp dan camp bersama keluarga”.

Hasil rekomendasi objek wisata: “Situs purbakala Gua Passea terletak di Desa Lembanna. Namanya diambil dari bahasa setempat yang berarti penderitaan. Objek wisata ini memiliki tipe horizontal dengan

rangkaian stalagtit dan stalagmit yang indah. Di dalamnya terdapat beberapa peninggalan berumur ratusan tahun. Konon, tempat ini dihuni oleh manusia sejak abad ke-8 sebelum masehi”.

1. Tahapan *Preprocessing*

Hasil Proses *Casefoling*

Objek wisata yang dicari: memiliki fasilitas toilet, penginapan, ruang shalat, tidak jauh dari pusat kota, akses jalan yang baik, tidak memakan biaya yang banyak dapat camp dan camp bersama keluarga

Hasil rekomendasi objek wisata: situs purbakala gua passea terletak di desa lembanna. namanya diambil dari bahasa setempat yang berarti penderitaan. objek wisata ini memiliki tipe horizontal dengan rangkaian stalagtit dan stalagmit yang indah. di dalamnya terdapat beberapa peninggalan berumur ratusan tahun. konon, tempat ini dihuni oleh manusia sejak abad ke-8 sebelum masehi.

Hasil *whitespace Insensitivity*

Objek wisata yang dicari: memilikifasilitas toilet penginapan ruangshalat tidak jauh dari pusat kota akses jalan yang baik tidak memakan biaya yang banyak dapat camp dan camp bersama keluarga

Hasil rekomendasi objek wisata: situs purbakala gua passea terletak di desa lembanna namanya diambil dari bahasa setempat yang berarti penderitaan objek wisata ini memiliki tipe horizontal dengan rangkaian stalagtit dan stalagmit yang indah di dalamnya terdapat beberapa peninggalan berumur ratusan tahun konon tempat ini dihuni oleh manusia sejak abad ke-8 sebelum masehi.

Pembentukan Rangkaian K-Gram (k=3)

Berikut hasil dari pembentukan rangkaian *K-Gram* dengan nilai k=3. Rangkaian *K-Gram* ini menghasilkan term-term dengan Panjang term sesuai dengan nilai Variabel k.

Table 1. Hasil rangkaian *K-Gram*

Rangkaian <i>K-Gram</i> objek wisata yang dicari	Rangkaian <i>K-Gram</i> hasil rekomendasi objek wisata
mem emi mil ili lik iki kif ifa fas asi sil ili lit ita tas ast sto toi oil ile let etp tpe pen eng ngi gin ina nap apa pan anr nru rua uan ang ngs gsh sha hal ala lat att tti tid ida dak akj kja jau auh uhd hda dar ari rip ipu pus usa sat atk tko kot ota taa aak aks kse ses esj sja jal ala lan any nya yan ang ngb gba bai aik ikt kti tid ida dak akm kme mem ema mak aka kan anb nbi bia iay aya yay aya yan ang ngb gba ban any nya yak akd kda dap apa pat atc tca cam amp mpd pda dan anc nca cam amp mpb pbe ber ers rsa sam ama mak ake kel elu lua uar arg rga	sit itu tus usp spu pur urb rba bak aka kal ala lag agu gua uap apa pas ass sse sea eat ate ter erl rle let eta tak akd kdi did ide des esa sal ale lem emb mba ban ann nna nan ana nam ama man any nya yad adi dia iam amb mbi bil ild lda dar ari rib iba bah aha has asa sas ase set ete tem emp mpa pat aty tya yan ang ngb gbe ber era rar art rti tip ipe pen end nde der eri rit ita taa aan ano nob obj bje jek ekw kwi wis isa sat ata tai ain ini nim ime mem emi mil ili lik iki kit iti tip ipe peh eho hor ori riz izo zon ont nta tal ald lde den eng nga gan anr nra ran ang ngk gka kai aia ian ans nst sta tal ala lag agt gti tit itd tda dan ans nst sta tal ala lag agm gmi mit ity tya yan ang ngi gin ind nda dah ahd hdi did ida dal ala lam amn mny nya yat ate ter erd rda dap apa pat atb tbe beb ebe ber era rap apa pap ape pen eni nin ing ngg gga gal ala lan anb nbe ber eru rum umu mur urr rra rat atu tus usa san ant nta tah ahu hun unk nko kon ono non ont nte tem emp mpa pat ati tin ini nid idi dih ihu hun uni nio iol ole leh ehm hma man anu nus usi sia ias ase sej eja jak aka kab aba bad adk dke ke8 e8s 8se seb ebe bel elu lum umm mma mas ase seh ehi

2. *Rolling Hash*

Setiap term-term yang didapatkan dari pembentukan rangkaian *K-Gram*, maka selanjutnya adalah menghitung nilai hash dari setiap term yang terbentuk menggunakan persamaan (1).

Rolling Hash dari “mem”

a) Nilai ASCII setiap karakter

Table 2. Table ASCII untuk term “mem”

Char	Dec
m	109
e	101
m	109

b) Nilai hash untuk term “mem”

$$H_{(mem)} = (c_1 \cdot b^{(k-1)} + c_1 \cdot b^{(k-1)} + \dots + c_{(k-1)} \cdot b^k + c_k) \text{ mod } q$$

$$H_{(mem)} = (ASCII(m) \cdot 7^2 + ASCII(e) \cdot 7^1 + ASCII(m) \cdot 7^0) \text{ mod } 5001$$

$$H_{(mem)} = (109 \cdot 7^2 + 101 \cdot 7^1 + 109 \cdot 7^0) \text{ mod } 5001$$

$$H_{(mem)} = (6157) \text{ mod } 5001$$

$$H_{(mem)} = 1156$$

berikut hasil dari perhitunga Rolling Hash terhadap rangkaian KGram objek wisata yang dicari dan hasil rekomendasi objek wisata.

Table 3. Nilai hash

Nilai hash objek wisata yang dicari	Nilai hash hasil rekomendasi objek wisata
1156 816 1183 1005 1133 998 1079 955 791	1485 1073 1617 1649 1535 1420 1628 1368 587
662 1477 1005 1142 1053 1477 673 1557 1565	598 1029 605 1073 590 962 1523 633 1281 672
1281 1001 1114 872 1568 1304 821 1215 891	1540 1438 743 665 1504 854 1442 1114 857
1011 1180 633 1276 636 1304 1501 1521 625	1469 601 1047 734 945 721 850 1421 609 1107
1225 955 1459 882 605 1086 680 1600 1518	809 1123 590 632 1256 1178 619 1177 612
941 685 607 1081 989 675 1560 892 692 655	1129 643 1333 1707 557 731 932 613 1131 644
1432 1045 1421 1634 1429 671 1543 1135 1347	1000 1088 692 655 1418 927 584 577 889 654
1459 538 616 1148 1456 859 1473 980 605	1428 658 1457 861 1499 823 1221 1282 685
1080 643 1333 1717 625 1208 829 585 594	1627 1717 625 1208 833 622 843 1378 666
1009 1159 1518 941 685 610 1106 1156 808	1502 1530 1029 1304 818 1190 720 851 1436
1126 598 1031 620 1180 633 944 696 1728 696	1053 1459 541 633 1264 1230 644 1007 816
1717 625 1208 829 590 643 1333 1714 601	1180 1680 1046 1429 661 1467 597 1019 1233
1039 690 633 1282 663 1473 638 627 1224	1008 1156 816 1183 1005 1133 998 1093 1061
1284 688 621 1179 638 627 1222 1274 622 861	1530 1029 1298 787 986 1341 1442 1109 1864
1487 1422 612 1126 602 1057 821 1207 1525	1324 1298 1470 608 1092 716 821 1207 835
653 1403	636 1284 1374 625 1217 892 1026 584 933 637
	1310 1543 1470 605 1073 589 963 1534 1056
	1480 688 637 1310 1543 1470 605 1073 582
	914 1191 1077 1627 1717 625 1215 891 1014
	1186 682 580 900 734 941 686 605 1079 625
	1231 1333 1723 665 1504 846 1382 690 633
	1282 662 1470 606 735 622 843 1376 633 1278
	637 1304 823 1234 1017 1213 864 833 605
	1080 620 1176 622 863 1513 1612 1273 1644
	1480 1380 681 1617 1634 1423 638 1298 1466
	597 1024 1609 1249 1129 1319 1276 1324 1302
	1499 823 1221 1282 669 1528 1019 1224 949
	738 989 1024 1607 1235 1029 1295 1102 785
	955 1129 639 1323 1642 1466 938 658 1447
	787 979 598 1019 535 580 559 749 1005 455
	3650 1439 735 616 821 1219 1604 1200 1134
	658 1445 781

3. Menghitung *Similarity*

Setelah melalui proses *Rolling Hash*, maka selanjutnya adalah menghitung nilai *similarity* dengan menggunakan persamaan (2).

$$S = ((2 * C) / (A + B)) * 100\%$$

$$S = ((2 * 72) / (140 + 290)) * 100\%$$

$$S = 33.49\%$$

Ket:

A : jumlah *term* pada inputan pencarian objek wisata

B : jumlah *term* pada hasil rekomendasi objek wisata

C : total *term* yang sama dari inputan pencarian objek wisata dan hasil rekomendasi objek wisata

Nilai *Similarity* yang diperoleh yaitu sebesar 33.49% merupakan hasil persentase kemiripan dari pencarian objek wisata dan hasil rekomendasi objek wisata yang memperoleh persentase kemiripan tertinggi dibandingkan dengan beberapa objek wisata lainnya. Nilai pada Variabel A dapat diperoleh melalui data pada Tabel 3, sedangkan nilai pada Variabel B dapat diperoleh melalui data pada Tabel 3 dan nilai dari Variabel C yaitu hasil dari total *term* yang sama dari Tabel 3.

IV. Kesimpulan dan saran

Setelah melakukan tahapan penelitian Penerapan Metode *String Matching* dengan Algoritma *Rabin-Karp* untuk merekomendasikan objek wisata di Kabupaten Bulukumba diperoleh kesimpulan bahwa Algoritma *Rabin-Karp* dapat digunakan dan diterapkan untuk mendapatkan rekomendasi Objek Wisata pada sistem pencarian. Nilai kemiripan antara teks yang sama tetapi tata Bahasa yang berbeda atau perubahan posisi beberapa teks tidak akan mempengaruhi nilai kemiripan atau *similarity*.

Penerapan Algoritma *Rabin-Karp* dalam sistem ini untuk mendapatkan rekomendasi objek wisata masih memiliki beberapa kekurangan karena keterbatasan. Untuk itu ada beberapa saran yang mungkin dapat digunakan dalam pengembangan sistem ini kedepannya yaitu sistem ini dapat dibangun dengan mencoba menerapkan metode atau algoritma lain baik itu dengan membandingkan Algoritma *Rabin-Karp* dengan metode lainnya atau mengkombinasikan dengan metode lainnya untuk mendapatkan rekomendasi objek wisata. Penelitian ini sudah dapat menentukan nilai kemiripan antara pencarian objek wisata dan rekomendasi objek wisata, tetapi belum ada bentuk pengujian untuk mengetahui nilai akurasi dari setiap hasil kemiripan yang dihasilkan oleh Algoritma *Rabin-Karp*. Sistem ini masih kurang dalam proses *preprocessing* seperti penghapusan kata tertentu (*stopword removal*).

Daftar Pustaka

- [1] A. Wahyuni and N. E. Risa, "Analisis Kesesuaian Kawasan Wisata Pantai Di Pantai Panrangluhu Kabupaten Bulukumba," *Agrominansia*, vol. 4, no. 2, pp. 61–75, 2020, doi: 10.34003/293020.
- [2] I. Ahmad, R. I. Borman, G. G. Caksana, and J. Fakhrurozi, "Implementasi String Matching Dengan Algoritma Boyer-Moore Untuk Menentukan Tingkat Kemiripan Pada Pengajuan Judul Skripsi/Ta Mahasiswa (Studi Kasus: Universitas Xyz)," *SINTECH (Science Inf. Technol. J.)*, vol. 4, no. 1, pp. 53–58, 2021, doi: 10.31598/sintechjournal.v4i1.699.
- [3] A. Filcha and M. Hayaty, "Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa," *JUITA J. Inform.*, vol. 7, no. 1, p. 25, 2019, doi: 10.30595/juita.v7i1.4063.
- [4] H. F. A. SINAGA, "Implementasi Algoritma Rabin Karp Pada Pendeteksian Pengulangan Kata Kerja Dalam Penulisan Bahasa Inggris," *skripsi*, pp. 7–37, 2017, [Online]. Available: <http://repositori.usu.ac.id/handle/123456789/4657>.
- [5] E. Ernawati, A. Johar, and S. Setiawan, "Implementasi Metode String Matching Untuk Pencarian Berita Utama Pada Portal Berita Berbasis Android (Studi Kasus: Harian Rakyat Bengkulu)," *Pseudocode*, vol. 6, no. 1, pp. 77–82, 2019, doi: 10.33369/pseudocode.6.1.77-82.
- [6] D. N. Sari and D. P. Utomo, "Implementasi Algoritma Rabin-Karp Pada Pencarian Quotes Tokoh Terkenal," *Pelita Inform. Inf. dan Inform.*, vol. 9, no. 1, pp. 43–55, 2020.
- [7] M. O. Pohan and R. D. Sianturi, "Penerapan Algoritma Rabin-Karp Pada Pencarian Sinonim Kata," *J. Informatics Manag. Inf. Technol.*, vol. 2, no. 1, pp. 12–17, 2022.
- [8] A. A. B. Ginting and D. P. Utomo, "Perancangan Aplikasi Catalog Wisata Di Sumatera Utara Menggunakan Algoritma Rabin-Karp," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3,

- no. 1, pp. 57–63, 2019, doi: 10.30865/komik.v3i1.1568.
- [9] E. Siswanto and Y. Ceng Giap, “Implementasi Algoritma Rabin-Karp dan Cosine Similarity untuk Pendeteksi Plagiarisme Pada Dokumen,” *J. Algor*, vol. 1, no. 2, pp. 16–22, 2020, [Online]. Available: <https://jurnal.buddhidharma.ac.id/index.php/algor/index>.
- [10] A. P. U. Siahaan and Sugianto, “Analisis k-gram, basis dan modulo rabin-karp sebagai penentu akurasi persentase kemiripan dokumen,” *SENASPRO 2017 / Semin. Nas. dan Gelar Prod.*, pp. 198–206, 2017.